



エージェント指向ユビキタスミドルウェアにおける 適応型プラットフォーム間通信機構

著者	伊藤 大視
学位授与機関	Tohoku University
URL	http://hdl.handle.net/10097/39881

平成 20 年度 修士学位論文

エージェント指向ユビキタスミドルウェア
における適応型プラットフォーム間通信機構

東北大学大学院情報科学研究科 情報基礎科学専攻

博士課程前期 2 年の課程

コミュニケーション論講座(白鳥研究室)

A7IM1004 伊藤 大視

目次

第1章 序論	1
1.1 研究の背景	1
1.1.1 ユビキタス情報環境とユビキタスアプリケーション	1
1.1.2 エージェント指向ミドルウェア	2
1.2 本研究の目的	3
1.3 本論文の章構成	4
第2章 関連研究と課題	5
2.1 ユビキタスアプリケーションとエージェント指向ミドルウェア	5
2.2 ユビキタス情報環境におけるエージェント指向ミドルウェアの課題	6
2.3 本研究で扱う問題点	7
第3章 提案: 適応型プラットフォーム間通信機構	13
3.1 提案の概要	13
3.2 適応型プラットフォーム間通信機構の設計	16
3.2.1 提案機構のアーキテクチャ	16
3.2.2 通信方式選択機能の内部構造	20
第4章 実装	22
4.1 実装	22

第 5 章 実験と評価	27
5.1 実験環境とシナリオ	27
5.2 実験結果	31
5.3 考察	40
第 6 章 結論	41
6.1 おわりに	41
謝辞	43
発表論文	44
参考文献	45

図目次

2.1 エージェント間通信とプラットフォーム間通信	7
2.2 コネクション型/単一セッション継続型のみの通信方式の場合	9
2.3 コネクションレス型のみの通信方式の場合	10
3.1 エージェント間通信の性質に応じたプラットフォーム間通信方式選択の例	15
3.2 適応型プラットフォーム間通信機構のアーキテクチャ	17
3.3 通信方式選択機能	20
4.1 DASH のプラットフォーム間通信部	23
4.2 適応型プラットフォーム間通信機構のプロトタイプシステム	24
5.1 実験のネットワーク及びソフトウェア構成	29
5.2 プロトタイプシステムにおけるプラットフォーム間通信	32
5.3 プロトタイプシステムにおける提案手法のプラットフォーム間通信の例 . .	33
5.4 実験 1(PHS 網を使用) による実験結果	34
5.5 実験 2(FOMA 網を使用) による実験結果	35

表 目 次

2.1	代表的なエージェント指向ミドルウェアのプラットフォーム間通信方式 . . .	11
2.2	プラットフォーム間通信方式とトランスポート層プロトコルの関係	11
2.3	エージェント間通信の性質	12
3.1	エージェント間通信	14
5.1	計算機構成	28
5.2	実験のネットワーク構成	28
5.3	実験に用いた送信データレート要求	30
5.4	実験 1 の応答時間	32
5.5	実験 2 の応答時間	33
5.6	実験 1 を 30 回の試行した結果	38
5.7	実験 2 を 30 回の試行した結果	39

第1章 序論

近年，計算機資源の拡充や計算機自体の小型化・省電力化の促進により，計算機の偏在性が高まりつつある [1]．このような計算機の普遍化による情報環境はパーベシブコンピューティングと呼ばれ様々な研究活動が行われている．一方，普遍化した計算機資源を活用する新たな情報環境としてユビキタス情報環境 [2, 3] への注目も高まっている．本研究はユビキタス情報環境上で動作するアプリケーション (ユビキタスアプリケーション) の構成技術であるエージェント指向ミドルウェアにおける通信効率の向上を目指すものである．

本章では，ユビキタスアプリケーションの特徴を説明し，その上でユビキタス情報環境でサービスを提供する場合の問題点について述べる．続いて本研究の目的と本研究による効果について簡単に述べ，最後に本論文の構成について述べる．

1.1 研究の背景

1.1.1 ユビキタス情報環境とユビキタスアプリケーション

ユビキタス情報環境上ではデバイス，ソフトウェアなどの様々な構成要素からなる即興的なサービスによって利用者にサービスを提供する．現在，構成要素の状況 (コンテキスト) に応じたサービス提供 (コンテキストウェアサービス) や，情報環境の構成要素が動的に連携しサービスを合成するサービス合成技術 [4, 5, 6, 7, 8] などに関する研究・開発が

盛んに行われている [9] . 従来の情報環境では利用可能な計算機資源・ネットワーク資源は有限であり, サービスを提供する場合にはその品質に応じた資源を使用するため, サービスの品質と情報環境上の資源にはトレードオフ関係が存在していた. 従来の情報環境におけるサービス提供では, このようなトレードオフに対して静的なシステム・デバイス構成による最適化が行われていたが, ユビキタス情報環境においてはサービスを構成する構成要素が環境・時間によって刻々と変化するため, 動的にシステムを構成, および再構成する必要がある. 従ってユビキタス情報環境上で動作するアプリケーション (ユビキタスアプリケーション) には, 多様な環境への適応能力や利用者がサービスを要求する環境や利用可能なデバイスの構成に対処する能力が求められる. このような性質のアプリケーションを効果的に開発するための基盤システムとして「ユビキタスアプリケーション向け基盤システム」に対する期待が高まりつつある [10] .

1.1.2 エージェント指向ミドルウェア

1.1.1 節 ではユビキタス情報環境におけるサービス提供が環境へ適応し動的にシステムを構成する必要があることを述べた. 一方, 近年, 環境適応能力および分散協調システムを構築するためのパラダイムとしてエージェント指向によるシステム設計が注目されてきている. 特に自律性・適応性・自発性・協調性などを備えた小規模のエージェントを組み合わせることで, 動的にシステムを構築するマルチエージェントシステムは, ユビキタスアプリケーションに求められる能力を十分に満たしていると考えられる事ができ, ユビキタスアプリケーション向けのミドルウェアとしてその期待が高まりつつある [12] . このような背景から, 様々なエージェント指向ミドルウェアの研究・開発が行われている [11, 13, 14, 15, 16, 17, 18, 19, 20] .

しかし, エージェント指向ミドルウェアをユビキタスアプリケーションの開発に応用す

る際に、エージェント間通信の効率低下によるサービスの安定性の低下や、計算機資源の制約によるパフォーマンス低下の問題、これらの問題を複合したシステムのスケーラビリティに関する問題、およびシステムの配備方法に関する複雑化など様々な問題が存在する。これらは、ユビキタス情報環境において計算能力の低い小型携帯端末・埋め込み型コンピュータや不安定な無線ネットワークなどが主要な構成要素となり、従来想定されていた環境よりも大きな制約のある環境でサービスを提供する必要があるという性質に起因する。

1.2 本研究の目的

本研究は 1.1.2 節 で述べた問題において、エージェント間通信の効率低下によるサービスの安定性の低下の解消に焦点を置き、環境の状況やエージェント間通信の性質を考慮したエージェントプラットフォームの通信基盤の高度化により、エージェント指向ミドルウェアによる、より安定したユビキタスサービスの提供を実現することを目的としている。その目的の達成のため、ネットワーク資源、計算機資源の状況、エージェント間の通信に対する要求に応じて、エージェントプラットフォーム間の通信方式を柔軟に選択することが可能な「適応型プラットフォーム間通信機構」を提案する。

この機構により、資源制約の大きいユビキタス情報環境において、エージェント間の様々な通信要求を可能な限り満たすエージェントプラットフォームが実現でき、これに基づいてマルチエージェントシステムとして構成されるユビキタスアプリケーションによる、より高度で安定したユビキタスサービスの提供が期待できる。

1.3 本論文の章構成

本論文の章構成は以下のとおりである．

2 章では，既存のエージェント指向ミドルウェアにおけるプラットフォーム間通信について述べる．また，本研究の目的の達成のために解決すべき技術的課題を提起し，本研究で扱う問題を明らかにする．

3 章では，本研究における目的を達成するための提案である，適応型プラットフォーム間通信機構について述べる．また，その具体的な内容として2 章で提起した技術的課題を解決するための手法について詳細を述べる．

4 章では，提案方式の有効性を評価するためのプロトタイプシステムについて述べる．

5 章では，4 章で述べたプロトタイプシステムを用いて行った実験について述べ，その結果の考察に基づき，提案方式の評価を行う．

最後に6 章では本研究の結論と今後の課題についてまとめる．

第2章 関連研究と課題

2.1 ユビキタスアプリケーションとエージェント指向ミドルウェア

ユビキタスアプリケーション向け基盤システムに関する研究が行われている．[10]では，ユビキタスアプリケーションに対する機能要件として以下の項目を挙げ，ユビキタスアプリケーションを開発するためのミドルウェアを提案している．

(R1) 動的に変化する環境に適応する能力

(R2) アプリケーションを動的に構成する能力

(R3) 情報を様々なアプリケーション間で共有する能力

一方，近年様々なエージェント指向ミドルウェアの研究・開発が行われている [11, 13, 14, 15, 16, 17, 18, 19, 20]．これらのエージェント指向ミドルウェアは，マルチエージェントシステムを構築するための基盤である．ここで，マルチエージェントシステムとは，アプリケーションの構成要素である個々のソフトウェアコンポーネントをエージェントとして動作させ，エージェントの移動性，組織構成能力，協調性などに基づいて構成されるシステムであり，エージェントの組み合わせによって様々な状況に適応可能なシステムを柔軟に構築することが可能となる点が大きな特徴である．従ってマルチエージェントシステムの特徴は，上記の (R1) から (R3) の 3 つの機能要件を十分に満たすと考えられ，ユビキ

タスアプリケーションをマルチエージェントシステムとして構築することが有効であると考えられる [21]。また，エージェント指向ミドルウェアをユビキタスアプリケーション開発のための基盤システムとして利用することにより，従来のユビキタスアプリケーション向けミドルウェアと比較して，より高度な機能を提供できる可能性がある。

2.2 ユビキタス情報環境におけるエージェント指向ミドルウェアの課題

従来のエージェント指向ミドルウェアをユビキタス情報環境に適用する際に，いくつかの課題が存在する。ここで想定するユビキタス情報環境は，128kbps の PHS から数十 Mbps 程度の無線ネットワーク環境と，PDA から家庭用 PC 程度の計算機から構成され，様々な機能・性能の構成要素が混在するコンピューティング環境である。このようなコンピューティング環境では，ユビキタスアプリケーションが低帯域のネットワークや計算能力の低い計算機を利用する場合が想定される。そこで，ネットワークや計算機の資源が不足し，サービス品質 (QoS) が劣化したり動作が不安定になった場合でも，可能な限り安定したサービスを継続的に提供できるエージェント指向ミドルウェアが必要である。具体的には，エージェント指向ミドルウェアをユビキタス情報環境に適用する際の課題として，以下の 4 点が挙げられる。

(P1) エージェント間通信の効率低下によるサービスの安定性の低下

(P2) 計算機資源の制約によるパフォーマンスの低下

(P3) (P1)，(P2) の複合的な問題によるシステムのスケーラビリティの低下

(P4) システムの配備方法の複雑化

(P1) は主にネットワーク資源の制約による課題であり，(P2) は計算機資源の制約による課題である．(P3) は (P1) と (P2) に基づくシステム全体としての機能拡張性，規模拡張性に関する課題であり，(P4) は異種コンピューティング環境上にいかにエージェントプラットフォームを展開するかといった運用上の課題である．

2.3 本研究で扱う問題点

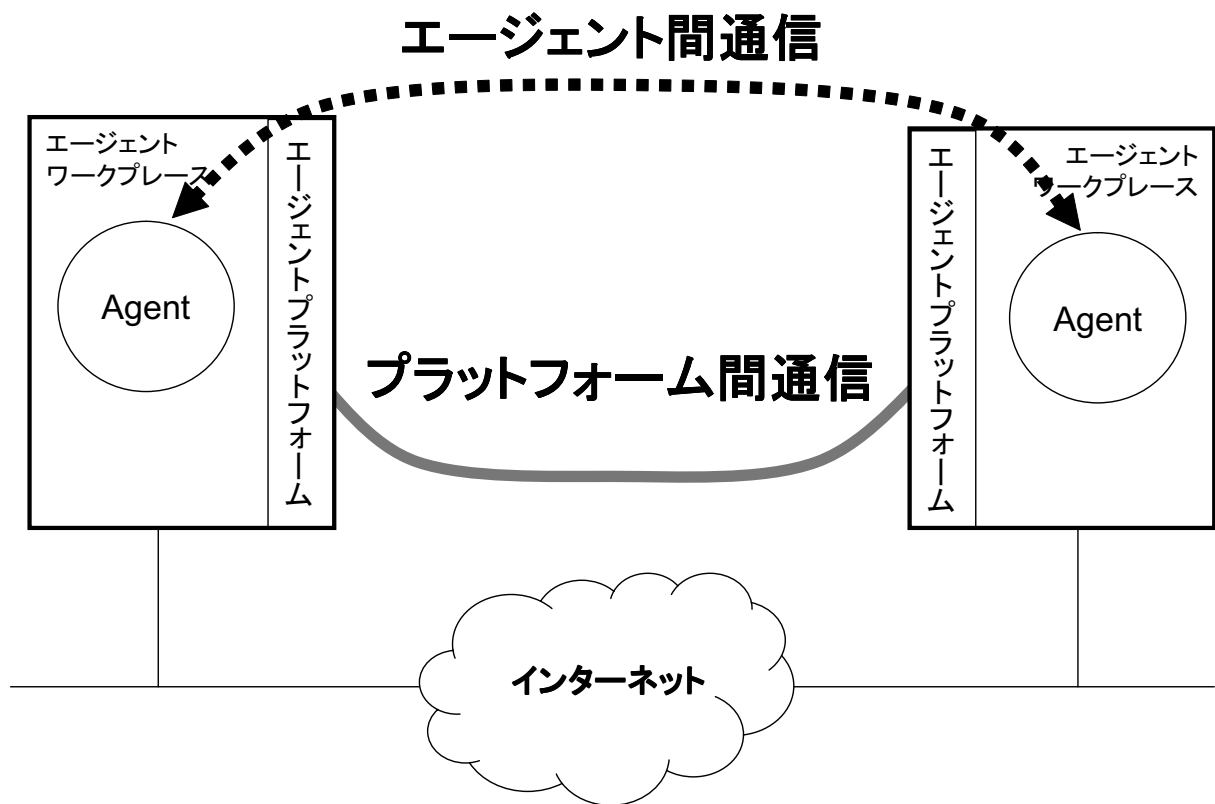


図 2.1: エージェント間通信とプラットフォーム間通信

本研究では，2.2 節で述べた課題のうち (P1) のエージェント間通信の効率低下によるサービスの安定性の低下の問題に焦点を当てる．エージェント間通信とは図 2.1 のようなエージェント間の様々な相互作用を伝播する“言語”としての役割を持つ通信であり，エージェント間通信は更にプラットフォーム間通信という下層の通信方式を用いて実現してい

る．ユビキタス情報環境では，ネットワーク帯域が狭く接続が不安定な状況が存在するため，エージェント間通信を支えるプラットフォーム間通信の効率の向上，および安定化が必須となる．しかし，現状のエージェント指向ミドルウェアではこの課題に十分に対応しているとは言い難い．これはプラットフォーム間通信がネットワーク環境の状況を考慮せず，単一の通信方式のみで行われるためである．そこで，[22] では様々なエージェントのシステム構成および通信方式の調査が行われており，その中での，主なエージェント指向ミドルウェアとそのプラットフォーム間通信方式の関係を表 2.1 に示す．またプラットフォーム間通信方式とトランスポート層プロトコルの関係を表 2.3 に示す．これらの表からは，主なエージェント指向ミドルウェアは様々なプラットフォーム間通信方式を利用可能なものの，そのトランスポート層プロトコルの通信方式では主に TCP を用いていることがわかる．また，[17] などはプラットフォーム間通信方式として Berkeley Socket を利用でき，様々なプラットフォーム間通信が可能だが，アプリケーションが指定する単一のトランスポート層プロトコルを用いている．これらのことから，主なエージェント指向ミドルウェアは単一の通信方式によるプラットフォーム間通信方式を用いていることがわかる．

ユビキタス情報環境において，エージェント指向ミドルウェアが単一の通信方式によるプラットフォーム間通信方式のみで通信を行う際に発生する問題を図 2.2 および図 2.3 に示す．図 2.2，図 2.3 は複数のエージェント指向ミドルウェアがインターネットを介して接続されている状況で，様々な計算機上のエージェントが，表 2.3 に示す通信を行っている．

まず，図 2.2 はプラットフォーム間の通信方式がコネクション型/単一セッション継続型のみで，その通信方式が TCP であった場合の例である．この通信方式を利用するエージェント指向ミドルウェアでは，Agent-7 と Agent-3 の「動作終了後の動作履歴送信」などのエージェント間で単発で確実な通信が必要な場合に，ある程度の帯域を使用できれば問題

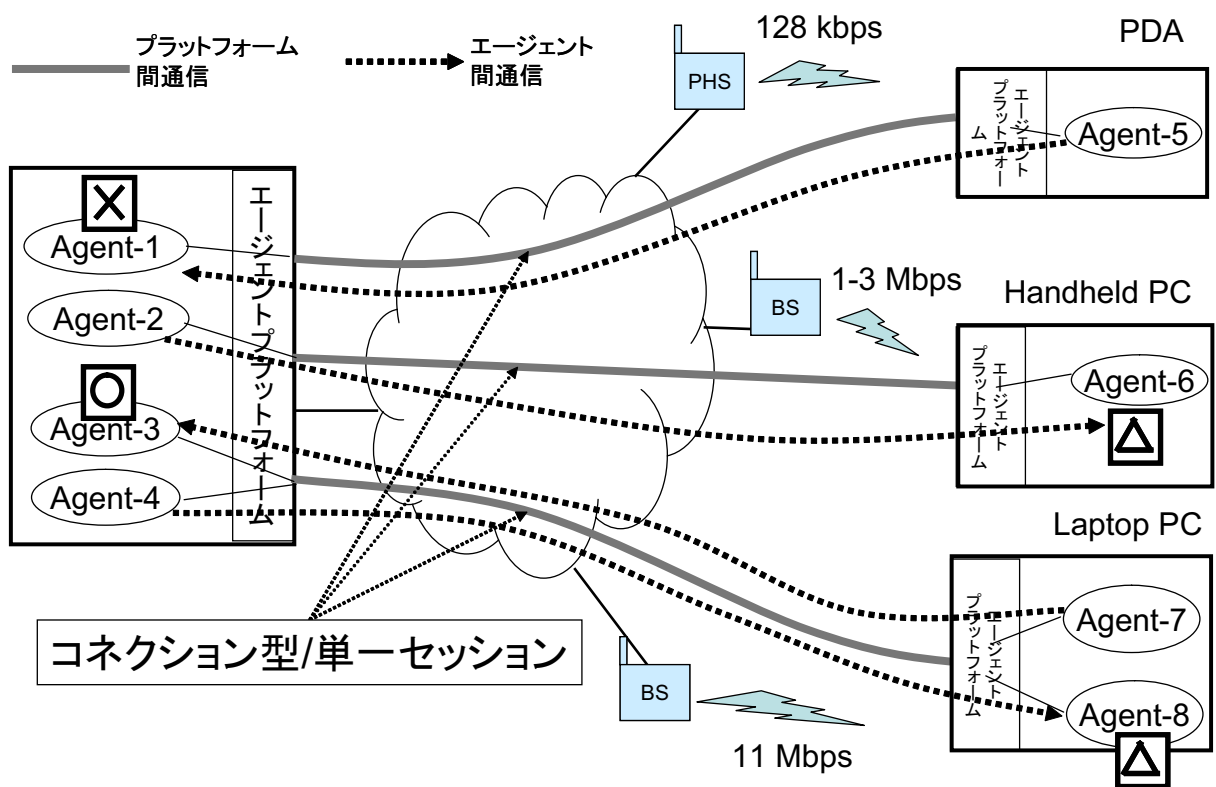


図 2.2: コネクション型/単一セッション継続型のみの通信方式の場合

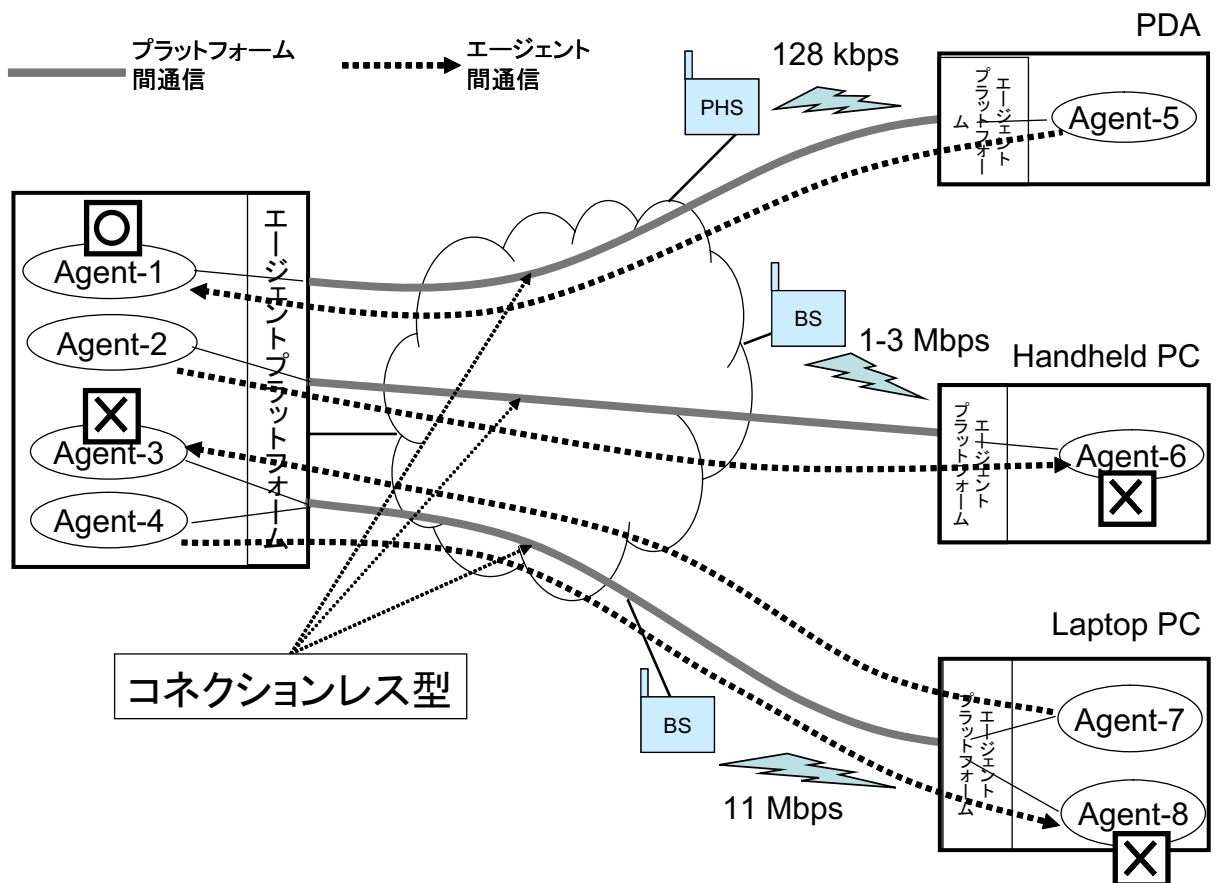


図 2.3: コネクションレス型のみの通信方式の場合

表 2.1: 代表的なエージェント指向ミドルウェアのプラットフォーム間通信方式

ミドルウェア名	開発元	プラットフォーム間通信方式
Ajanta[15]	University of Minnesota	ATP, XML, RMI
FIPA-OS[16]	Open Source	ACL, IIOP , RMI
Grasshopper[17]	IKV++ Technologies AG	sockets, RMI, IIOP
JADE[18]	TILab	ACL, XML, MTP(HTTP , WAP)
Zeus[19]	British Telecom. Lab	KQML(*2), ACL
DASH[14]	千葉工業大学	ACL, RMI

*1)ATP: Agent Transfer Protocol

*2)KQML: Knowledge Query and Manipulation Language

表 2.2: プラットフォーム間通信方式とトランスポート層プロトコルの関係

プラットフォーム間通信方式	トランスポート層プロトコル
RMI	トランスポート層プロトコルに非依存 (主に TCP)
HTTP, IIOP	TCP

なく通信が行われる。しかし，Agent-5 と Agent-1 の「位置情報の継続的報告」の様な定期的に情報を送信する場合など，リアルタイム性を必要とするようなエージェント間通信の場合は，確実性を保障するためにリアルタイム性が犠牲となるため，ネットワークが不安定な場合に大きな遅延が発生し，アプリケーションの通信要求を十分に満たすことが困難となる。次に，図 2.3 はプラットフォーム間の通信方式としてコネクションレス型通信のみを使用し，その通信方式が UDP であった場合の例である。このプラットフォーム間の通信方式の場合，UDP では輻輳制御が行われないため確実性よりもリアルタイム性を必要とするエージェント間通信に有効である。しかし，Agent-7 と Agent-3 の「動作終了後の動作履歴送信」のような個々のエージェントへの動作指示を送信する場合，確実性

を必要とする通信方式が必要となるため，例えば，メッセージの欠落によりエージェントの意図しない動作が発生し協調動作に問題が発生する可能性がある．このように，ネットワーク環境の状況を考慮せず単一の通信方式のみで行う既存手法では，ネットワーク資源や計算機資源の変化が大きいユビキタス情報環境において，エージェント間通信の効率の向上やサービスの安定化が困難であると言える．

表 2.3: エージェント間通信の性質

送信	受信	通信種類	エージェント間通信の性質
Agent-5	Agent-1	位置情報の継続的報告	リアルタイムな通信
Agent-2	Agent-6	エージェント動作終了指示	確実に遅延の少ない通信
Agent-7	Agent-3	動作終了後の動作履歴送信	確実な通信
Agent-4	Agent-8	コントラクトネットプロトコルのメッセージ	確実に遅延の少ない通信

第3章 提案: 適応型プラットフォーム間 通信機構

3.1 提案の概要

2.3 節で述べた問題を解決するためには、計算機の状態に応じて、プラットフォーム間で複数の通信方式を選択して利用することが可能な機構が必要である。ここで、計算機の状態とは計算機資源 (CPU 使用率、メモリ使用率、ディスプレイの解像度など) や、ネットワーク資源 (ネットワークの接続状況、利用可能なネットワークプロトコル、トラフィック量、エージェント個々の通信帯域など) の資源状況と、エージェント間通信の性質 (個々のエージェントの通信期間、通信内容の抽象的な意味、マルチエージェントによる組織構成全体の総合的な意味など) とする。これらの資源状況、およびエージェント間通信の性質を基に通信方式を選択して利用することにより、例えば、資源制約のあるネットワーク環境において、図 3.1 のようにリアルタイム性が要求されるエージェント間通信の場合は転送制御がなく遅延の少ない通信 (UDP 等) を使用し、確実性が重視される場合は転送制御のある通信 (TCP 等) を提供するなど、エージェント間通信の様々な要求を可能な限り満たす通信路を提供することのできるエージェントプラットフォームが実現できる。そこで、本研究ではエージェントプラットフォーム内の機構として、「適応型プラットフォーム間通信機構」を提案する。本機構は、ネットワーク資源の状況、計算機資源の状況、エージェント間通信の性質や要求に基づき、適応的にプラットフォーム間通信方式を選択・利用する機能を有する。この適応的なプラットフォーム間通信方式を M と表現し、以下の

5つのタプルにより構成する．

$$M = \langle c, a, b, p, s \rangle$$

ここで各要素は

c : コネクション形式 (コネクション型 , コネクションレス型等)

a : エージェント間通信のセッション長 (単発 , 継続等)

b : データ転送形式 (バルク転送 , ストリーミング転送等)

p : 優先度転送 (優先転送 , 通常転送等)

s : セキュリティ転送 (暗号化 , プレーン等)

と定義する．このような要素から構成される M を用いた例を図 3.1 に示す．また表 3.1 に図 3.1 のエージェント間通信を示す．図 3.1 の Agent-A と Agent-B のエージェント間で , 継続的で遅延の少ない通信を要求していた場合は M として \langle コネクションレス型 , 単発 , ストリーム , 通常 , プレーン \rangle が選択され , 本機構によりプラットフォーム間で UDP/ストリーム型伝送が選択される．また , この UDP/ストリーム型転送が選択された上で , 更に Agent-C と Agent-D のようにシステム終了後の動作履歴の送信など単発で信頼性の必要な通信を行う場合には , M として \langle コネクション型 , 単発 , ストリーム , 高優先 , プレーン

表 3.1: エージェント間通信

送信	受信	通信種類
Agent-A	Agent-B	位置情報の継続的報告
Agent-C	Agent-D	エージェント動作終了指示
Agent-E	Agent-F	動作終了後の動作履歴送信
Agent-G	Agent-H	コントラクトネットプロトコルのメッセージ

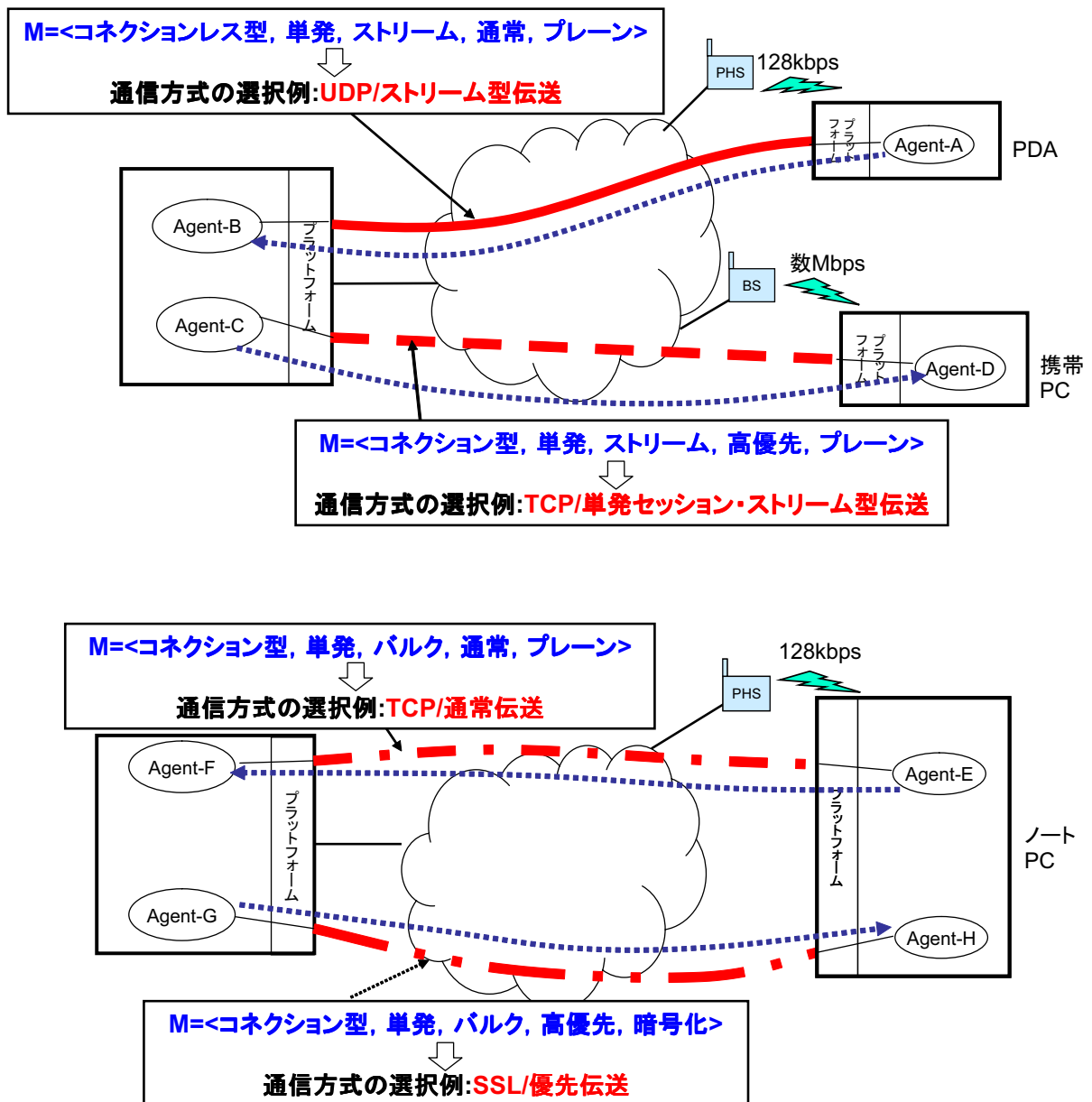


図 3.1: エージェント間通信の性質に応じたプラットフォーム間通信方式選択の例

ン> が選択され、本機構によりプラットフォーム間で TCP/単発セッション・ストリーム型伝送が選択される。また通信の頻度や通信の信頼性だけではなく、Agent-G と Agent-H のようにエージェント間で通信の秘匿化が必要な通信が要求された場合は、プラットフォーム間で SSL 通信などを選択することも可能となる。従って、本機構では M を用いて様々なエージェント間の通信の要求を柔軟に表現することで、 M に対してプラットフォーム間の通信方式を同時に対応することが可能となり、その結果、多様なエージェント間通信の要求に対応可能となる。

3.2 適応型プラットフォーム間通信機構の設計

3.2.1 提案機構のアーキテクチャ

図 3.2 に適応型プラットフォーム間通信機構のアーキテクチャを示す。計算機資源観測機能、ネットワーク資源観測機能、エージェント間通信観測機能の 3 つの観測機能と、これら観測機能より得られる観測情報を基に、プラットフォーム間通信方式 M を決定するための通信方式選択機能、またネットワークインタフェース (ネットワーク I/F) を通して、選択された M に基づき通信路を確保するプラットフォーム間通信制御機能の、5 つの機能から成る。各種観測機能は、様々なインタフェースにより直接的な値を形式化し、通信方式選択機能へ観測情報を報告する。本研究では、形式化に関して以下の様な再帰的な構造を持つ形式を検討している。

1. $(Item = v)$: 要素 $Item$ が値 v を持つ
2. $(Item.SubItem = v)$: 要素 $Item$ の子要素 $SubItem$ が値 v を持つ
3. $(Item = (v1, v2))$: 要素 $Item$ が複合要素 $(v1, v2)$ を持つ

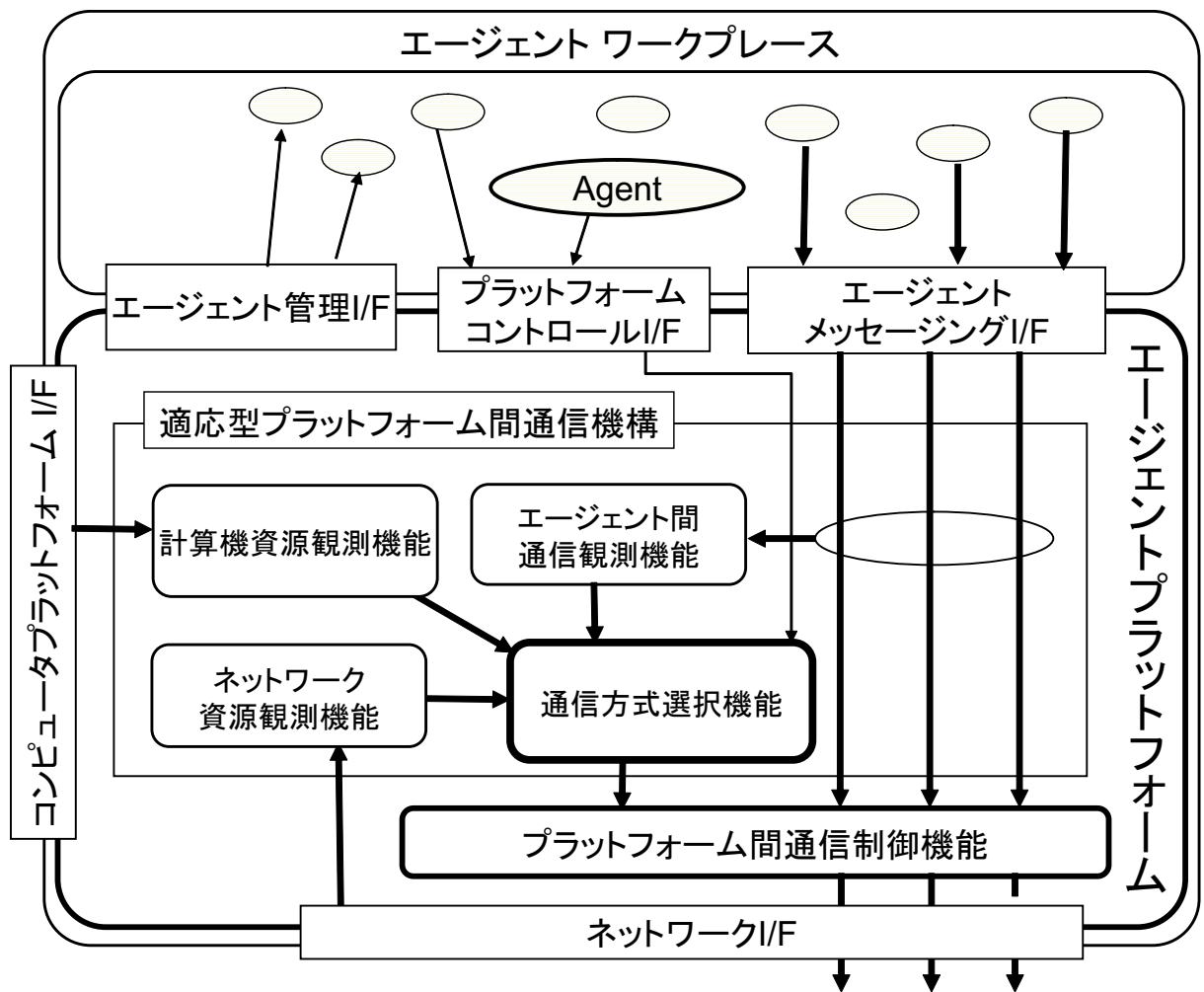


図 3.2: 適応型プラットフォーム間通信機構のアーキテクチャ

この 3 つの構造から成る表現形式を用いて観測機能の観測情報について述べる．

(1) 計算機資源観測機能

コンピュータプラットフォームインタフェース(コンピュータプラットフォーム I/F)を通してミドルウェアが動作する計算機の資源状況を観測する機能である．具体的な観測情報としては，CPU の使用率，利用可能メモリ，スクリーンなどの資源状況である．以下の式は，CPU の使用率が 15% であり，利用可能メモリが 200MB あり，スクリーンの色深度が 32bpp である資源状況を表している．

$$(CPU.Usage = 15\%, Memory.Available = 200MB, Screen.ColorDepth = 32bpp, \dots)$$

(2) ネットワーク資源観測機能

ネットワーク I/F を通して，計算機に備え付けられたネットワーク資源状況を観測する機能である．これは利用帯域や利用可能なネットワークプロトコル，ネットワークプロトコルに関する情報，トラフィック量などプラットフォーム間の通信に掛かる様々な情報が含まれ，以下のような形式で表現される．

$$\begin{aligned} & (\\ & \quad Network.Interface1.Bandwidth = 100Mbps, \\ & \quad Network.Interface1.Congestion = Yes, \\ & \quad Network.Interface1.IPAddress = (192.168.1.1, \dots), \\ & \quad Network.SupportedProtocol = (IPv4, TCP, UDP, SCTP, SSL, \dots), \\ & \quad Network.IPV4.property = (コネクションレス型, ロス率高, 運用性低, \dots), \\ & \quad Network.TCP.property = (コネクション型, ロス率低, コスト大, \dots), \\ & \quad Network.UDP.property = (コネクションレス型, ロス率中, コスト小, \dots), \\ & \quad \dots \end{aligned})$$

(3) エージェント間通信観測機能

ミドルウェア上に存在する，全てのエージェントのエージェント間通信を観測する機能である．エージェント間通信で送受信されるエージェントメッセージのヘッダやコンテンツ，エージェント間通信の頻度，送受信パターンなどからそのエージェントの性質を推測する．例えば一定間隔で通信を行い，かつ送信先のエージェントが常に同じである通信の場合は「このエージェントは位置情報などの何らかの定期的な資源状況の報告を行っている可能性がある」などという性質を推測する．

$$\begin{aligned} & (\quad \quad \quad AllAgents = (AgentA, AgentB, AgentC, \dots), \\ & \quad \quad \quad AgentA.Communication = (AgentX \text{ と通信, 継続的, 資源状況報告の可能性高}), \\ & \quad \quad \quad AgentB.Communication = (AgentX \text{ と通信, 断続的, 制御通信の可能性高}), \\ & \quad \quad \quad AgentC.Communication = (AgentY \text{ と通信, } \dots), \\ & \quad \quad \quad \dots \quad \quad \quad) \end{aligned}$$

(4) 通信方式選択機能

計算機資源観測機能，ネットワーク資源観測機能，エージェント間通信観測機能から得られた情報，およびプラットフォーム制御 I/F を介したエージェントからの明示的な指示に基づき，プラットフォーム間通信方式を決定する機能．この機能の内部の詳細は 3.2.2 節で後述する．

(5) プラットフォーム間通信制御機能

通信方式選択機能からの通知を受け，指定の通信方式に基づきプラットフォーム間通信路を確立し，特定のエージェント間メッセージと通信路との対応を行ってエージェントメッセージの転送を行う機能．また確立した通信路の継続・接続等の管理も行う．

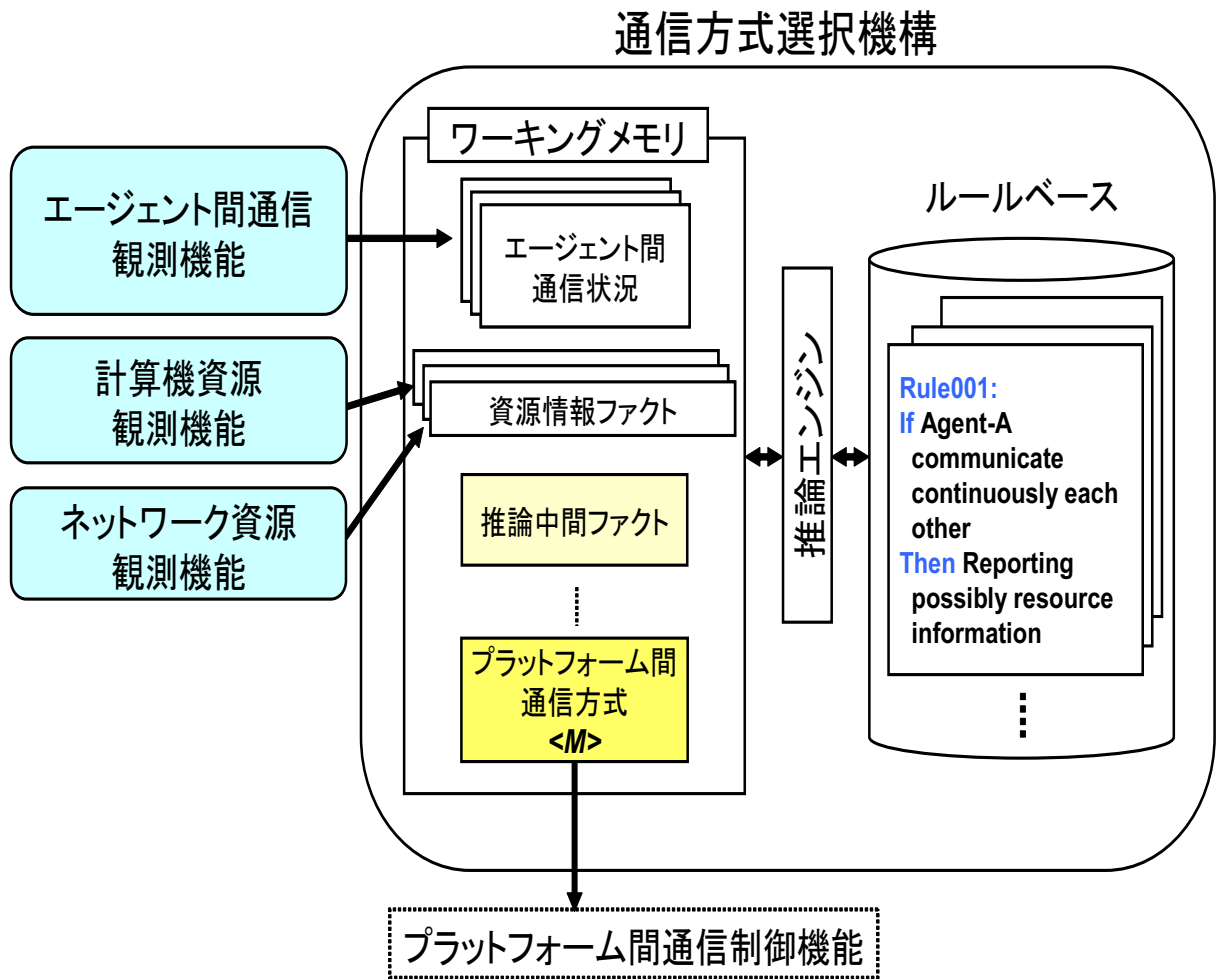


図 3.3: 通信方式選択機能

3.2.2 通信方式選択機能の内部構造

通信方式選択機能へ通知される各種観測機能の情報は多様であり，その中から通信方式を決定する方法を全てハードコーディングすることには限界がある．本研究ではこのような複雑な問題を解決するため，プロダクションシステムを用いることとする(図 3.3)．具体的には，まず各種観測機能より得られる状況をファクトとしてプロダクションシステムのワーキングメモリへ記憶させ，それらをプロダクションルールを用いて推論し，一時的な中間知識を導出する．中間知識の例としては，例えば計算機資源観測機能の $CPU.Usage = 15\%$ という観測情報から「暗号化のために使用可能な計算機資源がある」や，エージェント間

通信観測機能より，AgentYのエージェント間通信は制御通信を行っている可能性が高いという観測情報から「AgentYの制御通信のため暗号化が必要である」のようなものがある．これらの中間知識はそれぞれ異なりかつ競合する場合があるため，本研究ではエージェントアプリケーションプログラマにより動的にこのような競合を解消する知識を導入可能とする．そして，そのエージェントアプリケーション固有の競合解決手段を用いて，中間知識集合により最終的な通信方式を導出する過程を経て「適切な通信方式 M」を出力する．

第4章 実装

4.1 実装

3.2 節で述べた提案機構のアーキテクチャに基づき提案機構の実装を行った．実装にあたっては，エージェント指向ミドルウェアの一つである DASH[13] をベースソフトウェアとして選択し，そのエージェントプラットフォームに提案機構を組み込む形で実装した．ここで，DASH のプラットフォーム間通信部を図 4.1 に示す．

DASH においてプラットフォーム間通信を管理するプラットフォーム間通信制御機構は，プラットフォーム間通信方式として RMI を，通信方式として TCP を利用している．RMI はリモートのコンピュータにあるソフトウェアに組み込まれた関数を呼び出すためのフレームワークであり，関数呼び出しの際のパラメータに直列化に負荷がかかることや，関数の呼び出しから戻り値の取得までブロッキングを行うなど，様々なオーバーヘッドを生じるが確実な関数呼び出しが保障されている [23]．また，DASH のエージェントはプラットフォームに対して関連コンポーネントの読み込みや，エージェントの移動の要請などをプラットフォーム制御 I/F を用いて行っている．これまでに実装したプロトタイプシステムの概要を図 4.2 に示す．

プロトタイプシステムでは，まず DASH のプラットフォーム間通信制御機能を改良し，RMI ではなく，プラットフォーム間通信方式として TCP を利用するコネクション型通信，および通信方式として UDP を利用するコネクションレス型通信を平行して利用できるように拡張を行った．ここで実装上，単純な TCP ではプロトコル層でパケットがバッファ

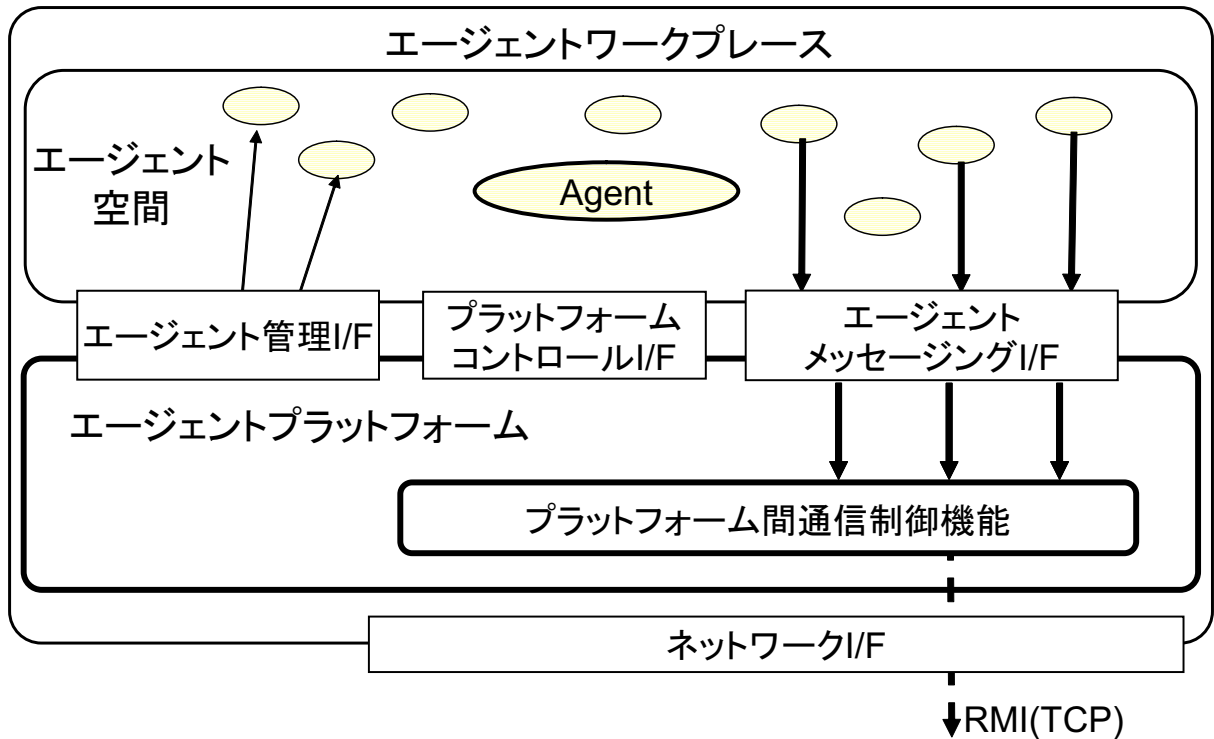


図 4.1: DASH のプラットフォーム間通信部

され、ネットワークの変化に敏速に応答できないため、「正常に送られたかどうか」を示す応答パケットを受信側が返信することとした。そこで、応答パケットの到着が一定時間以上遅くなるとネットワークの輻輳が大きくなったことを判断できるようにした。また、この応答パケットは1回のコネクション型通信につき1つ発生させ、応答時間がある閾値を越えると切断状態になり、一定時間を経て再接続を行うこととした。今回、このコネクション型通信の接続状態と切断状態の監視はネットワーク資源観測機能が行うものとした。また、コネクション型通信における応答パケットの許容時間、および再接続までの待機時間はプロトタイプシステムのシステムプロパティを通して動的に変更可能である。

次に、通信方式選択機能の実装とプラットフォームコントロールI/Fの改良を行い、エージェントにより通信の要求を通知可能にした。今回実装した通信方式選択機能は、エージェントがプラットフォームコントロールI/Fを通して通知した通信の要求と、ネットワー

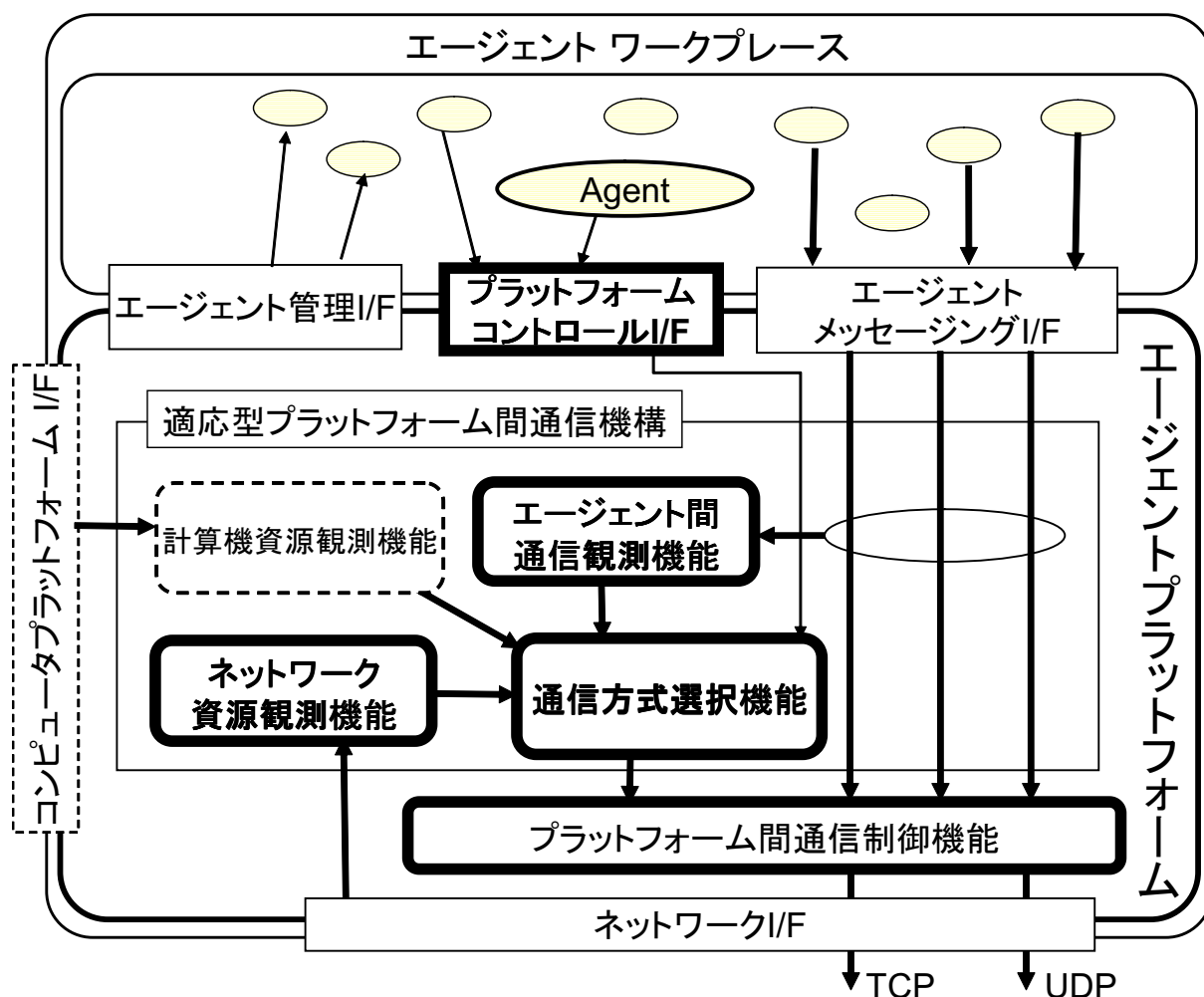


図 4.2: 適応型プラットフォーム間通信機構のプロトタイプシステム

ク資源観測機能によるコネクション型通信の接続状態と切断状態の監視から、コネクション型通信が接続状態の場合はエージェントの通信の要求を優先し、コネクション型通信が切断状態の場合はコネクションレス型通信に切り替える知識を静的に与えた。コネクション型が切断状態にコネクションレス型通信に切り替える知識は「TCP/IP ネットワークにおいてはフェアネスが考慮され、輻輳状況では輻輳制御のないUDPが優先的に送信される」という現実の状況から得られたものである。

これらのプロトタイプシステムの通信方式選択機能の知識を、以下のようなIF-THEN形式のルールで形式化を行った。

1. TCP/IP ネットワークでは輻輳状況において輻輳制御のない UDP が優先的に送信されるためコネクションレス型通信を優先する知識

IF *Network.TCP.Property* が ネットワーク輻輳 を含む

THEN *PlatformCommunication* へ

(NetworkRequirement, Connectionless – oriented) を追加

2. ネットワークの実状況による推論で得られる結果を優先する知識

IF *PlatformCommunication* が *(NetworkRequirement, ?x)* を含む

and PlatformCommunication が *(AgentRequirement, ?y)* を含む

THEN *PlatformCommunication* へ *(NetworkRequirement, ?x)* を設定

3. ネットワーク資源観測機能より得られた観測結果からプラットフォーム間通信方式を推論するための知識

IF *InadequacyCommunication* が *Connection – oriented* を含む

THEN *PlatformCommunication* へ

(NetworkRequirement, Connectionless – oriented) を追加

(?x や?y は IF-THEN ルールの IF 節で適合した任意の要素を意味する)

この通信方式選択機能の知識に対し，例えば「エージェントの通信の要求がコネクション型通信で，ネットワーク資源観測機能がコネクション型通信の切断状態を観測」という状況が観測された場合を考える．まず，通信方式選択機能へ次の 2 つのファクトが記憶される．

1. 「エージェントの通信の要求がコネクション型通信」より

PlatformCommunication へ *(AgentRequirement, Connection – oriented)* を追加

2. 「ネットワーク資源観測機能がコネクション型通信の切断状態を観測」より

InadequacyCommunication へ *Connection – oriented* を追加

続いて通信方式選択機能の知識より *InadequacyCommunication* に *Connection-oriented* が含まれるため, $PlatformCommunication = ((AgentRequirement, Connection-oriented), (NetworkRequirement, Connectionless-oriented))$ が推論される. ここでプラットフォーム間通信方式を現す *PlatformCommunication* で *Connection-oriented* と *Connectionless-oriented* とで衝突が起こっているため, さらに通信方式選択機能の「ネットワークの実況による推論で得られる知識を優先する知識」を用い $PlatformCommunication = ((NetworkRequirement, Connectionless-oriented))$ が得られる, そこでこれ以上の推論は行うことができないため, 通信方式選択機能はプラットフォーム間通信方式 M として *Connectionless-oriented*, つまり「コネクションレス型通信」を決定する.

第5章 実験と評価

5.1 実験環境とシナリオ

本稿で提案した適応型プラットフォーム間通信機構の有効性を確認するため、4.1 節で述べたプロトタイプシステムを用い、ネットワーク過負荷状況における提案手法の有効性を検証した。

実験環境としては、ユビキタス情報環境において即応性を必要とする通信の要求が発生するアプリケーションシナリオを想定した。具体的には、ユビキタス情報環境として計算機資源が豊富な PC と小型携帯端末機を、広域無線アクセス網を介して接続し、これらの計算機構成上で、動画像の送信データレートをコントロールすることが可能なストリーミング配信サービスを想定した。

図 5.1 に実験環境を示す。実験では、計算機資源が豊富な PC をストリーミング配信サーバとして、小型携帯端末機を受信携帯端末として使用した。また、表 5.2 に示す 2 種類のアクセスネットワークを用いた。実験 1 では(株)ウィルコム の PHS 網を利用する“b モバイル”を用いた (以降 PHS 網と略記する)。このアクセスネットワークは通信規格として“(株)ウィルコム 128kbps パケット通信方式”を用いており、上りの通信帯域幅が 68 kbps、下りの通信帯域幅が最大 32×4 kbps(128 kbps) のベストエフォートのネットワークである。実験 2 では(株)エヌ・ティ・ティ・ドコモの FOMA 網を利用する、(株)ウィルコム の“b モバイル 3G”を用いた (以降 FOMA 網と略記する)。このアクセスネットワークは、通信規格として HSPA(High Speed Packet Access) を用い、上りの通信帯域幅が 384 kbps、

下りの通信帯域幅が 3.6Mbps のベストエフォート方式のネットワークである．また，これらのアクセスネットワークでは，電波状況の悪化やネットワーク輻輳などが原因でネットワーク状況が容易に不安定となり，通信データの破棄や遅延に加えて遅延時間の揺らぎなどが起こるため，継続的でリアルタイムなサービスの制御が困難になるという問題を本提案機構によって解決する．次に，実験で用いられるエージェントのシステム構成を説明する．ストリーミング配信は UDP を用いて送信され，ストリーミング配信サーバ側の Transmitter エージェントが動画像を想定した計算機資源への負荷が少ないデータ(ダミーデータ)を受信携帯端末側の Receiver エージェントへ順次配信する．また，小型携帯端末機の Controller エージェントはエージェント間通信を用いてストリーミング配信サーバ側の Manager エージェントへ向け，定期的に動画像の送信データレートのコントロールを表すエージェントメッセージである“送信データレート要求”を送信する．Manager エージェントは，受信した送信データレート要求に従って，ストリーミング配信の送信データレートを変更する．

表 5.1: 計算機構成

	OS	CPU	メインメモリ
ストリーミング配信サーバ	Windows XP Professional(SP3)	1.4 GHz	480 MB
受信携帯端末	Windows XP Professional(SP3)	900 MHz	248 MB

表 5.2: 実験のネットワーク構成

	サービス名	通信規格	通信帯域幅(上り)	通信帯域幅(下り)	サービス品質
実験 1	(株)ウィルコム bモバイル	(株)ウィルコム 128kbps パケット通信方式	32 × 4 kbps(最大)	68 kbps	ベストエフォート
実験 2	(株)ウィルコム bモバイル 3G	HSPA	3.6 Mbps	384 kbps	ベストエフォート

また，実験で用いられる計算機の構成を表 5.1 に示す．そして，この実験環境上でプロトタイプシステムを用い，実験 1 および実験 2 で，従来手法として TCP のみの固定的な

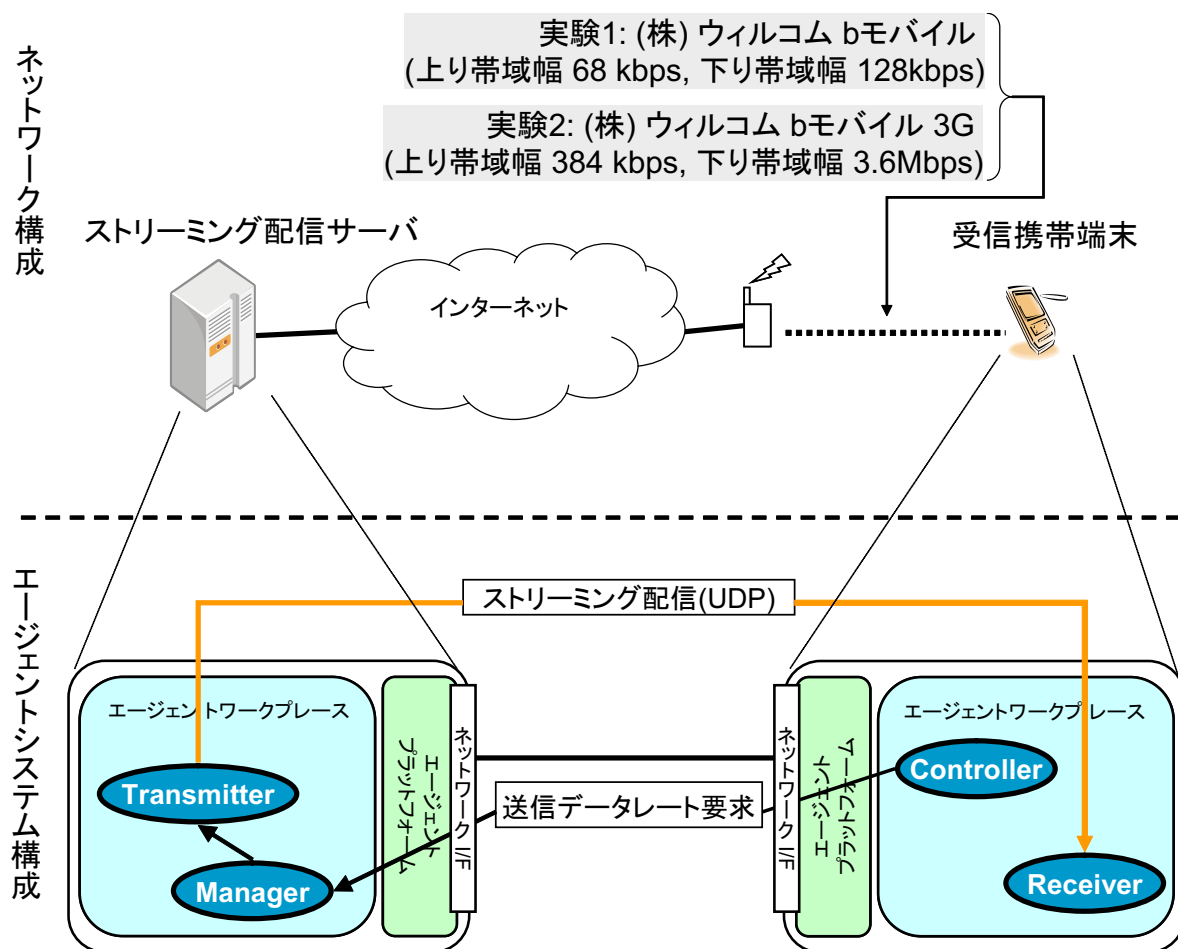


図 5.1: 実験のネットワーク及びソフトウェア構成

通信方式を使いエージェントメッセージを送信する手法，および提案手法としてネットワーク過負荷状況に応じて適応的に通信方式を選択しエージェントメッセージを送信する手法の 2 つの手法の比較を行った (図 5.2) .

本実験の目的は“ネットワーク過負荷状況における提案手法の有効性の検証”であり，ストリーミング配信のコントロールに要する時間である応答時間を計測する．この応答時間の比較により，ネットワーク状況が容易に不安定な状態になるユビキタス情報環境での有効性の確認を行う．また，実験で使用するパラメータとして，コネクション型通信における応答パケットの許容時間を実験 1 では 0.5 秒，実験 2 では 1 秒とし，コネクション型通信における再接続までの待機時間を 30 秒とした．実験 1 および実験 2 の送信データレ

ト要求の送信間隔は 5 秒である．また，送信データレート要求の値を表 5.3 に示す．また，表 5.3 の送信データレート要求の値は 1 周期分を表わし，繰り返して送られるものとする．

表 5.3: 実験に用いた送信データレート要求

	送信データレート要求の値
実験 1	[10, 30, 60, 100, 200, 400] kbps
実験 2	[100, 200, 240] kbps

実験 1 では約 60 秒間，実験 2 では約 45 秒間実験を行った．また説明のために図 5.2 にプロトタイプシステムにおけるエージェント間通信とプラットフォーム間通信のモジュールを示す．各実験において，従来手法では図 5.2 (a) のように通信方式選択機能による適応的な通信方式の推論を行わず，常にコネクションレス型通信を選択する．一方，提案手法は図 5.2 (b) のように通信方式選択機能を用い 4.1 節で述べた知識によって推論が行われ，コネクション型通信 (TCP) またはコネクションレス型通信 (UDP) を選択する．

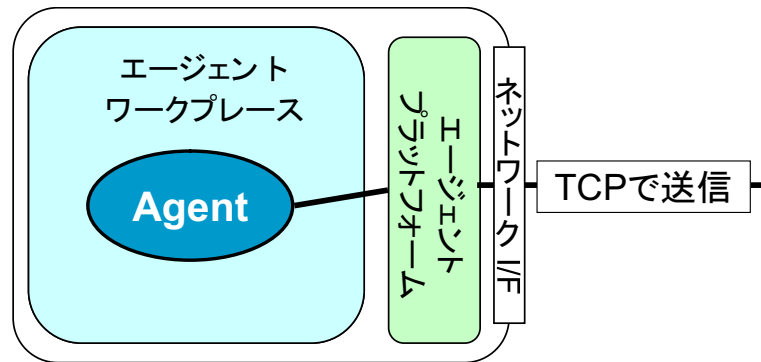
提案手法の具体的な動作例を図 5.3 を用いて説明する．ネットワークに輻輳が発生していない初期の段階では受信端末から送信データレート要求の送信は 4.1 節の知識により，エージェントの通信の要求であるコネクション型が優先され TCP を用いて送信される．一方，ネットワークに輻輳が発生しコネクション型通信において応答パケットの往復時間がコネクション型通信における応答パケットの許容時間を越えると，一旦，コネクション型通信が切断状態になりエージェント間通信の情報が破棄される．このコネクション型通信が切断状態となると即座にネットワーク資源観測機能によりその切断状態が観測され，4.1 節での「ネットワーク資源観測機能より得られた観測結果からプラットフォーム間通信方式を推論」の知識によりコネクションレス型通信が推論される．一方，エージェントの通信の要求は常にコネクション型通信であるため通信方式の推論で中間知識の衝突が発生するが，4.1 節の「ネットワークの実状況による推論で得られる結果を優先する」とい

う衝突を回避する知識により，コネクションレス型が選択される．この結果，図 5.3 の“送信データレート要求 (200 kbps)”はコネクションレス型通信で再送される．

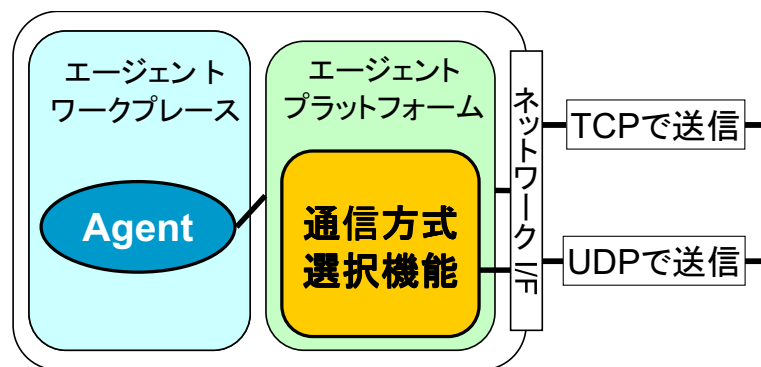
5.2 実験結果

実験 1(PHS 網を用いた実験)の実験結果を図 5.4 および応答時間の集計を表 5.4 に，また実験 2(FOMA 網を用いた実験)の実験結果を図 5.5 および応答時間の集計を表 5.5 に示す．それぞれのグラフの横軸は経過時間 (秒)，縦軸はデータレート (bps) を表しており，破線は受信携帯端末の Controller エージェントから送信された送信データレート要求の要求値を表し，実線は受信携帯端末の Receiver エージェントが受信したストリーミング配信データのデータレートを表している．またグラフ中の (A) や (B) などのアルファベットは送信データレート要求が変更される時点を表しており，(A') や (B') などは対応する送信データレート要求が受信データレートに反映された時点を示す．この (A) と (A') などの，送信データレート要求が変更される時点と受信データレートに反映された時点の時間差から応答時間を算出した．

まず図 5.4 (a) の実験 1 の従来手法による実験結果より，平均の応答時間は 2.2 秒であった．受信データレートは 20 秒時点までは送信データレート要求に対して安定していることがわかる．しかし表 5.4 (a) から，26.2 秒時点および 39.0 秒時点の応答時間が平均の応答時間の 2.5 倍から 4 倍程度長くなっている．これは 26.2 秒時点の送信データレート要求が 400kbps と，実験 1 のアクセスネットワークの下りの通信帯域幅である 128kbps より大きく，多量のストリーミング配信データによりネットワークで輻輳が発生し，また，コネクション型通信 (TCP) を使用して送信データレート要求を送信しているために，TCP の輻輳制御により送信データレートの要求がストリーミング配信サーバへ到達するのに時間が掛かったためである．このコネクション型通信の輻輳制御による送信データレート要求



(a) 従来手法



(b) 提案手法

図 5.2: プロトタイプシステムにおけるプラットフォーム間通信

表 5.4: 実験 1 の応答時間

(a) 従来手法の応答時間

送信データレート要求の送信時刻(秒)	1.4	5.3	10.5	15.7	20.9	26.2	39.0	53.6	55.5	56.1	56.9
応答時間(秒)	0.5	0.7	1.5	1.3	1.1	5.9	9.1	0.5	0.7	1.1	1.3

(b) 提案手法の応答時間

送信データレート要求の送信時刻(秒)	0.4	5.4	10.6	15.8	21.0	26.3	31.5	36.8	41.9	47	52.3	57.5
応答時間(秒)	1.5	0.5	1.3	1.1	2.0	4.7	(NA)	2.2	1.1	1.0	1.8	4.6

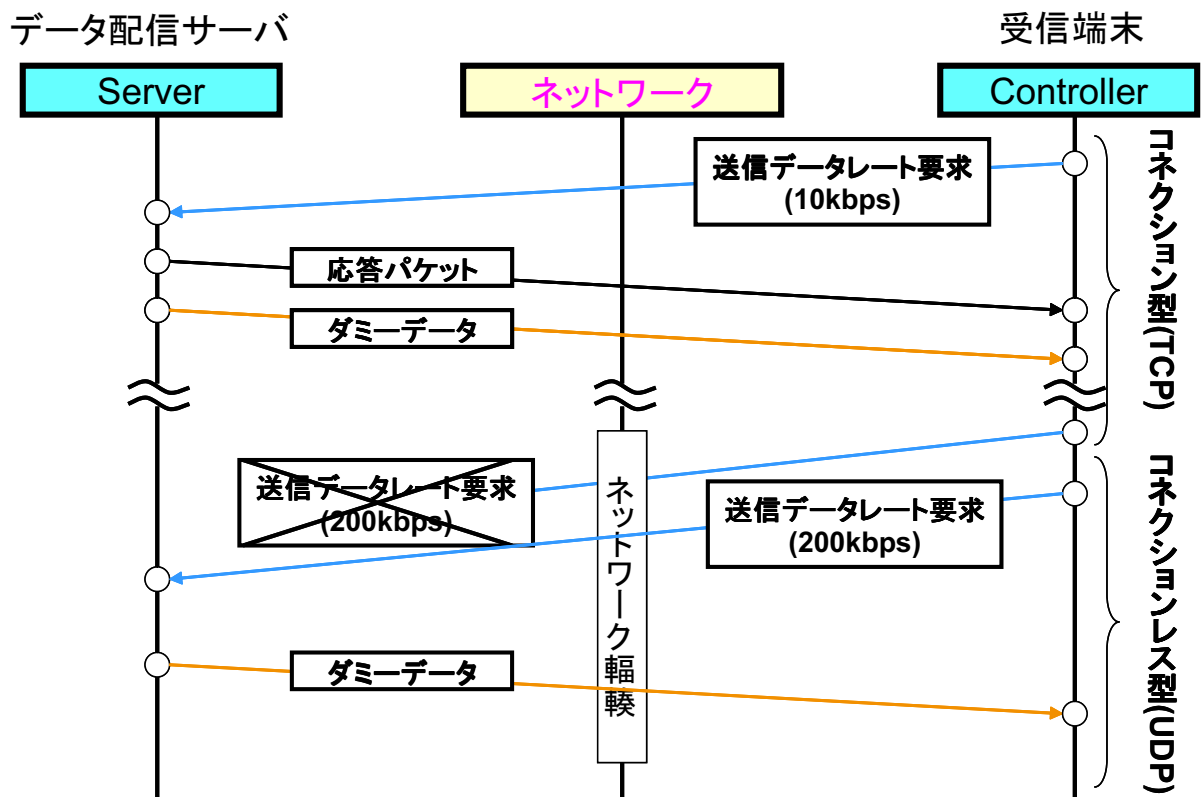


図 5.3: プロトタイプシステムにおける提案手法のプラットフォーム間通信の例

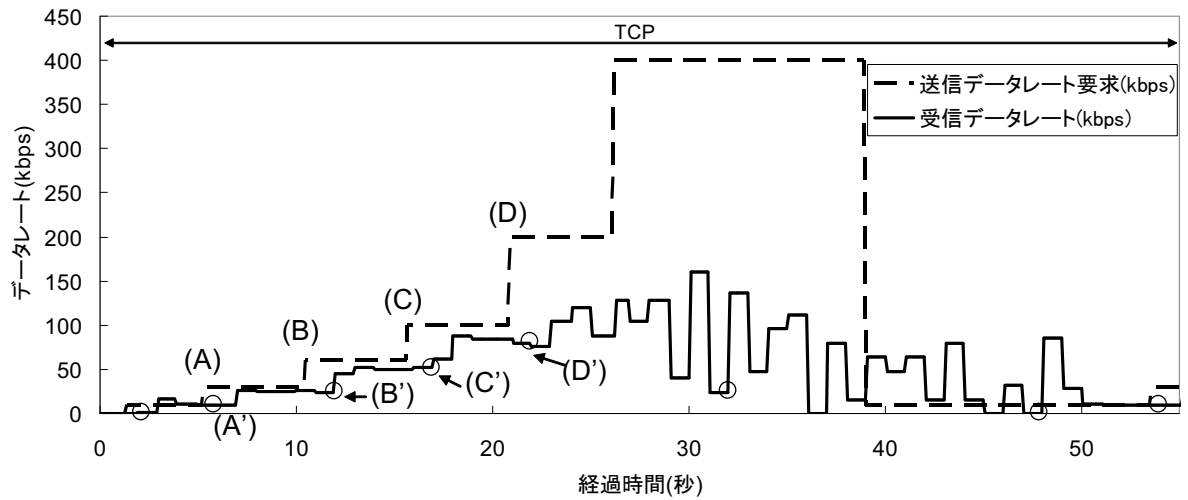
表 5.5: 実験 2 の応答時間

(a) 従来手法の応答時間

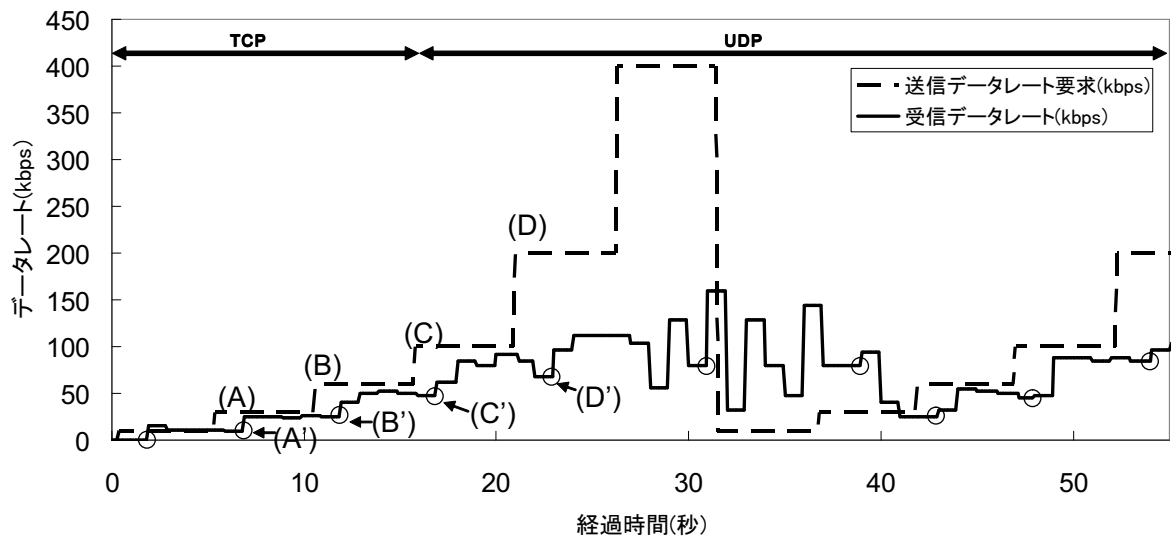
送信データレート要求の送信時刻(秒)	0.6	5.4	10.6	15.9	24.1	31.9
応答時間(秒)	1.4	1.5	3.3	9.1	7.9	9.1

(b) 提案手法の応答時間

送信データレート要求の送信時刻(秒)	0.7	5.2	10.4	15.7	20.9	26.0	31.3	36.5
応答時間(秒)	1.2	0.7	1.5	2.2	1.1	3.0	7.8	4.6

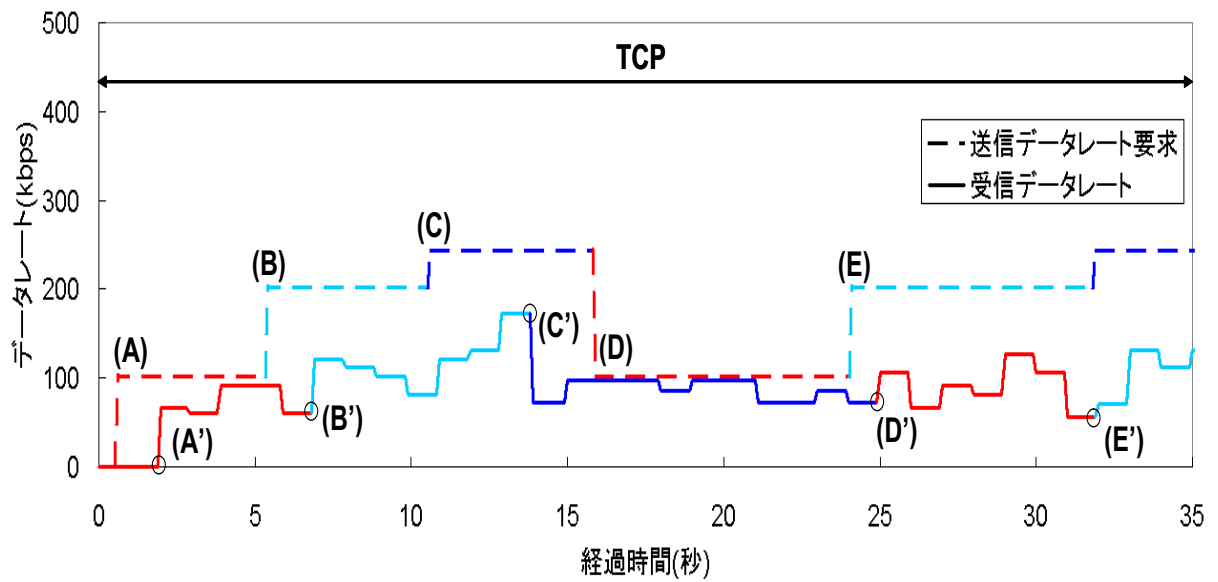


(a) TCP のみの固定的な通信方式による実験結果

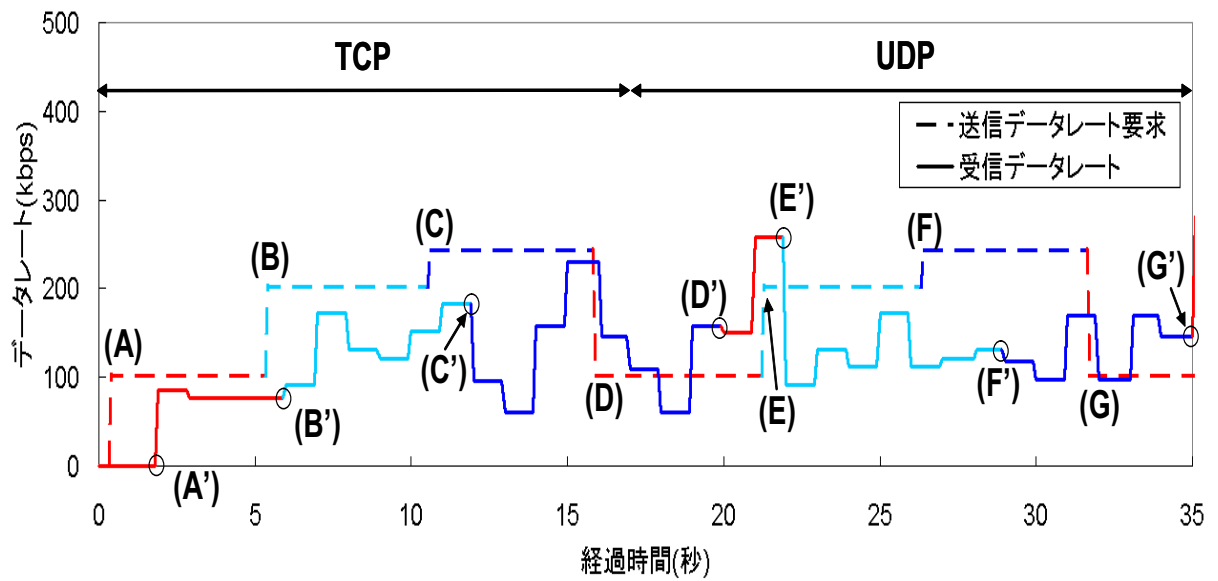


(b) 提案手法による実験結果

図 5.4: 実験 1(PHS 網を使用) による実験結果



(a) TCP のみの固定的な通信方式による実験結果



(b) 提案手法による実験結果

図 5.5: 実験 2(FOMA 網を使用) による実験結果

の到着の遅延は，39.0 秒時点で送信された 10kbps を要求する送信データレート要求がストリーミング配信サーバへ到着するまで続くため，39.0 秒時点でも依然としてネットワークに多くのストリーミング配信データが存在し輻輳状態となっている．このため 10kbps を要求する送信データレート要求の反映には時間がかかり，安定したストリーミング配信サービスを維持するためのエージェント間通信がストリーミング配信サービスの通信によって遅延していることがわかる．次に，図 5.4 (b) の提案手法の結果より，平均の応答時間は 2.0 秒であった．また約 10 秒時点で通信方式がコネクション型通信 (TCP) からコネクションレス型通信 (UDP) への切り替えが発生していた．これより提案機構一部であるネットワーク観測機能および通信方式選択機能が有効に働いていることが確認できる．表 5.4 (b) より，26.3 秒時点で応答時間が平均の応答時間の 2.4 倍長くなっており，さらに直後の 31.5 秒時点の送信データレート要求は応答時間に反映されていないことがわかる．26.3 秒時点の送信データレート要求の反映の遅れであるが，これは従来手法の結果と同様にストリーミング配信データによりネットワークで輻輳が発生した結果であると考えられる．また 31.5 秒時点の送信データレート要求はそのネットワーク輻輳により失われたと考えられる．しかし，36.8 秒の送信データレート要求は TCP/IP ネットワークにおけるフェアネスの考慮によりネットワークが輻輳状態でありながらも従来手法より早くストリーミング配信サーバへ到達した．その結果，ネットワーク輻輳が解消され，提案手法の実験では受信データレートが従来手法より安定化している．

続いて，図 5.5 (a) の従来手法による実験結果より，平均の応答時間は 5.4 秒であった．受信データレートは約 5 秒まで安定して送信データレート要求に追従しているが，それ以降の受信データレートは送信データレート要求に対して半分程度のデータレートでしか受信していないことがわかる．また，表 5.5 から 15.9 秒，24.1 秒，31.9 秒時点の応答時間が平均の 1.4 倍から 1.7 倍ということから，アクセスネットワークの 2% 程度の帯域使用

率でネットワーク輻輳が発生し、送信データレート要求のストリーミング配信サーバへの到着の遅延が起こっていることがわかる。また図 5.5 (b) の提案手法による実験結果での応答時間は 1.8 秒であった。更に受信データレートが送信データレート要求に追従しなくなり始める約 7 秒から、10.4 秒の送信データレート要求に対して応答時間が 1.4 秒、15.7 秒の送信データレート要求に対して 4.1 秒と長くなっていることがわかる。その後 17 秒時点でネットワーク輻輳が観測され通信方式が切り替えられた結果、従来手法に比べ 20.9 秒時点の送信データレート要求が 0.7 秒と短くなっている。しかし、この 20.9 秒時点の送信データレート要求は 100kbps を要求しており、かつ実験 2 の従来手法の結果からネットワーク輻輳は 100kbps の要求で常に発生していることから、この送信データレート要求の反映によってネットワーク輻輳の改善は起こらず、送信データレート要求の反映は早まっているものの 20 秒以降の受信データレートが不安定であることがわかる。FOMA 網は下り公称帯域が 3.6Mbps であるが、本実験結果では 100kbps 程度で輻輳が発生し、受信データレートが不安定になる現象が観測された。この状況は、FOMA 網内にて回線利用状況に応じて下り利用帯域に制限がかけられていることにより発生しているものと考えられる。一方、上りの回線については利用帯域の制限がかけられておらず、通信性能が比較的安定していたため送信データレート要求が効率的に伝送され、その結果エージェント間通信の効率が向上したと考えられる。

最後に実験 1 を 30 回試行した結果を表 5.6 に、実験 2 を 30 回試行した結果を表 5.7 に示す。表 5.6 より、従来手法での平均の応答時間は 2.3 秒であり、提案手法では 2.1 秒であることがわかる。また個々の実験の 1 回の試行における、応答時間の標準偏差の平均を取ると、従来手法で 2.3、提案手法では 1.6 と応答時間の標準偏差は提案手法が平均してより小さいことがわかった。これらの結果より、提案手法は従来手法に比べ「ネットワーク状況によらずエージェント間通信が影響されにくく、サービスの応答が一定になる」とい

表 5.6: 実験 1 を 30 回の試行した結果

従来手法			提案手法		
回	平均応答時間(秒)	応答時間の標準偏差	回	平均応答時間(秒)	応答時間の標準偏差
1	2.3	2.2	1	3.3	2.8
2	2.8	3.1	2	2.0	1.3
3	3.3	3.3	3	2.0	1.6
4	2.3	2.1	4	2.0	1.6
5	2.3	2.4	5	1.9	1.4
6	2.9	3.1	6	2.1	1.6
7	2.4	2.2	7	1.9	1.4
8	2.4	2.0	8	2.0	1.2
9	2.8	2.7	9	2.0	1.6
10	2.2	2.3	10	2.3	1.9
11	2.9	2.3	11	2.0	1.6
12	2.6	2.7	12	2.0	1.4
13	2.5	2.5	13	1.9	1.4
14	2.2	1.9	14	2.1	1.5
15	1.8	1.7	15	2.4	1.8
16	2.1	2.0	16	2.0	1.4
17	2.3	2.2	17	2.0	1.6
18	1.9	2.2	18	2.0	1.6
19	2.1	1.9	19	2.1	1.6
20	2.2	1.9	20	1.9	1.4
21	1.9	1.8	21	2.0	1.6
22	1.5	1.1	22	2.8	2.4
23	2.2	2.6	23	2.5	2.1
24	2.3	2.2	24	2.0	1.6
25	2.3	2.1	25	2.0	1.3
26	2.6	2.7	26	2.7	2.4
27	2.3	2.2	27	1.9	1.4
28	2.2	2.0	28	2.0	1.6
29	2.2	2.2	29	2.0	1.5
30	2.4	2.8	30	2.1	1.5
平均	2.3	2.3	平均	2.1	1.6

表 5.7: 実験 2 を 30 回の試行した結果

従来手法			提案手法		
回	平均応答時間(秒)	応答時間の標準偏差	回	平均応答時間(秒)	応答時間の標準偏差
1	3.9	12.1	1	2.8	5.0
2	4.7	11.8	2	2.5	2.9
3	5.4	11.5	3	1.5	1.8
4	4.7	11.0	4	2.5	1.7
5	4.8	10.1	5	2.6	1.5
6	4.7	9.9	6	1.8	1.4
7	4.6	9.4	7	1.7	0.5
8	4.4	9.3	8	1.3	0.2
9	4.9	9.2	9	1.2	0.2
10	3.2	6.3	10	1.1	0.1
11	5.2	6.0	11	1.0	0.1
12	2.3	3.8	12	1.0	0.1
13	1.6	1.3	13	0.9	0.1
14	1.6	0.5	14	0.9	0.1
15	1.0	0.3	15	1.1	0.1
16	1.3	0.2	16	1.0	0.1
17	1.2	0.2	17	0.9	0.1
18	1.0	0.1	18	0.9	0.1
19	1.1	0.1	19	0.9	0.1
20	1.1	0.1	20	1.0	0.1
21	1.0	0.1	21	1.1	0.1
22	1.1	0.1	22	0.9	0.1
23	1.0	0.1	23	0.8	0.1
24	1.0	0.1	24	1.0	0.1
25	1.0	0.1	25	1.0	0.1
26	1.1	0.1	26	1.0	0.1
27	0.9	0.1	27	1.1	0.1
28	1.0	0.1	28	0.8	0.1
29	1.1	0.1	29	1.0	0.1
30	0.9	0.1	30	1.0	0.1
平均	2.4	3.8	平均	1.3	0.6

うことがわかる．これは表 5.7 から確認することができ，従来手法の平均の応答時間は従来手法で 2.4 秒，提案手法で 1.3 秒であった．また応答時間の標準偏差の平均は従来手法で 3.8，提案手法で 0.6 であった．よって PHS 網および FOMA 網のいずれのアクセスネットワークでも，従来手法に比べ本提案手法はサービスの応答時間の短縮によってサービスの安定性を実現し，その結果サービス品質の向上が可能となったことを確認した．

5.3 考察

今回の実験では送信データレート要求が高データレートを要求しストリーミング配信データがネットワークを圧迫してネットワーク過負荷状況に陥った場合に，本提案手法がネットワーク輻輳を観測し，通信方式選択機能の知識を基に適応的に通信方式を選択した．その結果，送信データレート要求の反映を早め，ストリーミング配信の制御が不能になる時間を短縮したことを確認した．更に実験を 30 回試行した結果より，提案手法が従来手法に比べ短い応答時間で安定したストリーミング配信の制御を行っていることを確認した．よって実験目的である“ネットワーク過負荷状況における提案手法の有効性の検証”を達成したと言える．また実験ではネットワーク過負荷状況においてエージェント間通信の効率低下による送信データレート要求の遅延が発生し，ストリーミング配信サービスの制御不能な状態を短縮できた．このことから本研究の課題である (P1) エージェント間通信の効率低下によるサービスの安定性の低下を解消できたと言える．

第6章 結論

6.1 おわりに

本研究はユビキタス情報環境上で動作するユビキタスアプリケーションの構成技術であるエージェント指向ミドルウェアにおける通信効率の向上を目指すものであり、エージェント間通信の要求を満たすエージェント指向ミドルウェアによる、より安定したユビキタスサービスの実現を目的としている。従来のエージェント指向ミドルウェアではエージェントの通信の要求が様々であるのに対し、単一の通信方式によるプラットフォーム間通信を行っている。そのため、ネットワーク資源に制約があり、変化の大きいユビキタス情報環境におけるエージェント間通信の効率低下、サービスの不安定化の問題に対して、エージェント指向ミドルウェアが計算機の状態に応じて、プラットフォーム間で複数の通信方式を利用することの出来る機構である「適応型プラットフォーム間通信機構」を提案した。適応型プラットフォーム間通信機構は、計算機の状態として計算機資源やネットワーク資源といった資源状況とエージェント間通信の性質を基に、エージェント間通信とプラットフォーム間の通信方式を同時に対応させることで、同時に多様なエージェント間通信の要求に対応することが可能となる機構である。また、提案機構の一部を実装したプロトタイプシステムを用いアクセスネットワークとして PHS 網と FOMA 網を用いた検証実験を行った。実験の結果より、ネットワーク過負荷状況における提案手法の有効性を確認し、更に本研究の課題である (P1) エージェント間通信の効率低下によるサービスの安定性の低下の問題を解決した。

今後は、資源状況の正確な観測や、エージェントの通信の要求の抽象的な記述方法の設計など更なる提案機構の強化を行い、有効性の検証を推し進める予定である。また、通信方式選択機能の通信方式を選択する知識の妥当性の検証を行う予定である。

謝辞

本論文を終えるにあたり，このような興味深い研究テーマを与えていただき，日頃の研究活動の際に貴重な御指導を賜りました東北大学電気通信研究所教授・白鳥則郎先生に心から感謝致します．

研究に指導にご多忙な中，本論文を審査していただきました，東北大学サイバーサイエンスセンター教授・曽根秀昭先生に深く感謝いたします．

また，東北大学サイバーサイエンスセンター教授・木下哲男先生には，審査時にとどまらず，大学院セミナー等を通し，日頃から数多くの有益な御指導，御助言を賜りました．心より感謝いたします．

そして，東北大学電気通信研究所准教授・菅沼拓夫先生には，本研究をはじめ研究生生活全般にわたって終始御指導，御助言を賜りました．お忙しい折にも夜遅くまで相談に応じていただき，時には励ましていただきました．心より感謝いたします．

最後に，研究生生活の内外で私を支えてくれた多くの友人と，これまで精神的，経済的に支えてくれた両親に深く感謝して，本論文を締めくくりたいと思います．

発表論文

国際会議

1. T. Ito, S. Imai, H. Takahashi, T. Suganuma , and N. Shiratori, “Adaptive Inter-platform Communication Mechanism in Agent-based Middleware for Ubiquitous Computing Environments,” *Proc. of the 4th International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2008)*, pp.25–32, Jun. 2008.
2. T. Ito, H. Takahashi, T. Suganuma, and N. Shiratori, “Evaluation of Adaptive Inter-platform Communication Mechanism for Ubiquitous Multiagent Systems,” *The 3rd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS2009)* Mar. 2009. (accepted)

研究会・ワークショップ

1. 伊藤 大視, 高橋 秀幸, 菅沼 拓夫, 白鳥 則郎, “ユビキタス環境向けエージェントプラットフォームのための適応型通信機構の設計,” 信学技報, Vol. 108, No. 218, MoMuC2008-55, pp. 75–80, 2008.
2. T. Ito, H. Takahashi, T. Suganuma, and N. Shiratori, “Design and Implementation of Adaptive Communication Mechanism for Ubiquitous Multiagent Systems,” 第 16 回マル

チメディア通信と分散処理ワークショップ論文集, Vol. 2008, No. 14, pp. 187–192,
Dec. 2008. (最優秀論文賞受賞)

3. 伊藤 大視, 今井 信太郎, 高橋 秀幸, 菅沼 拓夫, 白鳥 則郎, “ユビキタス環境向けエージェント指向ミドルウェアにおけるプラットフォーム間通信機構,” 情報ネットワーク9月研究会 (IN), Sep. 2007.

(口頭発表)

参考文献

- [1] 小林 真輔, 越塚 登, 坂村 健, “4. 超小型チップネットワーキング技術 (小特集: ユビキタスネットワーク技術開発プロジェクト),” 電子情報通信学会誌, Vol. 91, No. 7, pp. 595–603, Jul. 2008.
- [2] M. Weiser, “The Computer for the Twenty-first Century,Scientific American,” Vol. 265, No. 3, pp. 94–104, 1991.
- [3] 田原 康生, “1. はじめに : ユビキタスネット社会の実現に向けて (小特集: ユビキタスネットワーク技術開発プロジェクト),” 電子情報通信学会誌, Vol. 91, No. 7, pp. 563–568, Jul. 2008.
- [4] 武本 充治, 大石 哲矢, 岩田 哲弥, 山登 庸次, 田中 洋平, 徳元 誠一, 島本 憲夫, 黒川 章, 須永 宏, 小柳 恵一, “ユビキタスコンピューティング環境に適したサービス提供アーキテクチャにおけるサービス合成方式とその実装,” 情報処理学会論文誌, Vol. 46, No. 2, pp. 418–433, 2005.
- [5] M. Inoue, K. Mahmud, H. Murakami, M. Hasegawa, and H. Morikawa, “Context-Based Network and Application Management on Seamless Networking Platform,” Wireless Personal Communications, Vol. 35, No. 1-2, pp. 53–70, Oct. 2005.
- [6] M. Minoh and T. Kamae, “Networked Appliance and Their Peer-to-Peer Architecture AMIDEN,” *IEEE Comm. Magazine*, Vol. 39, No. 10, pp. 80–84, 2001.

- [7] S. Gribble, M. Welsh, R. Behren, E. Brewer, D. Culler, N. Borisov, S. Czerwinski, R. Gummadi, J. Hill, A. Joseph, R. Katz, Z. Mao, S. Ross, and B. Zhao, “The Ninja architecture for robust Internet-scale systems and services,” *Special Issue of Computer Networks on Pervasive Computing*, Vol. 35, No. 4, pp. 473–497, Mar. 2001.
- [8] R. Masuoka, B. Parsia, and Y. Labrou, “Task Computing - The Semantic Web Meets Pervasive Computing,” *Proc. of the 2nd International Semantic Web Conference (ISWC2003)*, LNCS 2870, pp. 866–881, 2003.
- [9] 山本 修一郎, “ユビキタス・サービス・アーキテクチャ (連載: ユビキタス社会を支える IT 技術),” *ビジネスコミュニケーション*, Vol. 41, No. 8, 2004.
- [10] R. Grimm, J. Davis, E. Lemar, A. Macbeth, S. Swanson, T. Anderson, B. Bershad, G. Borriello, S. Gribble, and D. Wetherall, “System support for pervasive applications,” *ACM Transaction on Computer Systems (TOCS)*, Vol. 22, No. 4, pp. 421–486, Nov. 2004
- [11] 川村 隆浩, 田原 康之, 長谷川 哲夫, 大須賀 昭彦, 本位田 真一, “Bee-gent: 移動型仲介エージェントによる既存システムの柔軟な活用を目的としたマルチエージェントフレームワーク,” *電子情報通信学会論文誌*, Vol. J82-D-I, No. 9, pp. 1165–1180, 1999.
- [12] 鄭 顕志, 中村 博之, 川俣 洋次郎, 吉岡 信和, 深澤 良彰, 本位田 真一, “ユビキタスコンピューティングにおけるアプリケーション開発手法に関する研究動向,” *コンピュータソフトウェア*, Vol. 25, No. 4, pp. 121–132, 2008.
- [13] S. Fujita, H. Hara, K. Sugawara, T. Kinoshita, and N. Shiratori, “Agent-based design model of adaptive distributed systems,” *The International Journal of Artificial Intelligence, Neural Networks and Complex Problem-Solving Technologies*, Vol. 9, No. 1, pp. 57–70, 1998.

- [14] <http://www.agent-town.com/dash/index.html>
- [15] <http://www.cs.umn.edu/Ajanta/>
- [16] <http://fipa-os.sourceforge.net/>
- [17] <http://www.grasshopper.de/>
- [18] <http://jade.cselt.it/>
- [19] <http://www.opensource.org/>
- [20] <http://cougaar.org/>
- [21] 鄭 顕志, 石川 冬樹, 吉岡 信和, 深澤 良彰, 本位田 真一, “適応エージェントのための
ユビキタスコンピューティングミドルウェア,” コンピュータソフトウェア, Vol. 25,
No. 4, pp. 11–19, 2008.
- [22] G.T. Nguyen, T.T. Dang, L. Hluchy, M. Laclavik, Z. Balogh, and I. Budinska, “Agent
Platform Evaluation and Comparison,” II-SAS, Pellucid EU 5FP IST-2001-34519 RTD,
Technical report Jun 2002, Bratislava, Slovakia.
- [23] <http://java.sun.com/j2se/1.5/pdf/rmi-spec-1.5.0.pdf>
- [24] <http://www.bmobile.ne.jp/personal/base/128kbps.html>